Amendments To The Drawings:

The attached drawing sheets include changes to FIG. 3F. These sheets contain corrections shown in red for the examiner's approval and are requested to replace the original sheets of FIG. 3F.

Attachment: Replacement Sheet of FIG. 3F

Annotated Sheet Showing Changes of FIG. 3F

REMARKS/ARGUMENTS

Reconsideration is respectfully requested.

Claims 1-13 are pending before this amendment. By the present amendment, claims 1-13 are amended. No new matter has been added.

In the office action (page 2), the drawings stand objected to for including reference characters (142, 250, 310, 312, 314, 330, 350, 356, 360, 362, 364, 366, 368, 370 and 372) not mentioned in the description. In response, the applicants have amended the specification to include these reference characters, and FIG. 3F has been amended changing the second reference character 350 to reference character 358. Withdrawal of the objection is respectfully requested.

In the office action (page 2), the disclosure stands objected to for containing informalities. In response, the specification has been amended to remove the informality in the disclosure. Accordingly, withdrawal of the objection is respectfully requested.

In the office action (page 3), claims 8-9 and 12 stand rejected under 35 U.S.C. § 112, ¶2 as being indefinite. The examiner states that the words "concurrently" and "simultaneously" are unclear, "since it is not explained in the specification how the memory is able to write and delete parts of the memory at exactly the same time. The applicants have determined these words unnecessary, and in order to broaden the scope of claims 8-9 and 12, the words "concurrently" and "simultaneously" have been replaced with "together". As such, claims 8-9 and 12 no longer require memory to write and delete at *exactly* the same time. Instead, the writing and deleting is merely done

together. Support for the amendments to claims 8-9 and 12 can be found in the specification page 10, lines 1-13 and page 11, line 19 to page 12, line 12.

In the office action (page 4), claims 1-3, 10 and 13 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,640,529 (Hasbun). The "et al." suffix is omitted in a reference name.

The applicants respectfully **disagree**.

The present invention relates to a garbage collection method that balances the load of garbage collection by distributing and performing the garbage collection in a plurality of communication cycles. By dividing the load of the garbage collection, the presently claimed invention reduces the possibility of a response delay or a timeout in the command/response communication environment of a smart cart.

In the presently claimed invention, when garbage collection is performed over a plurality of the communication cycles (as in FIGS. 3C-3F), an address **list** of objects is stored in memory, and the sweep phase (where objects are deleted from memory) of garbage collection is performed using the address list (specification page 9, lines 1-6). As such, it is clear that the list of the presently claimed invention is an actual **list of objects to be deleted**.

Further, the presently claimed invention, a residual time is calculated after an external command is processed, and objects are deleted from the list within the calculated residual time. If objects remain on the list, the list is updated to list those undeleted objects, and the list is stored as described above.

Claim 1 was amended to clarify the presently claimed invention. Claim 1 as amended removes any confusion as to the list of the presently claimed invention. The

list of the presently claimed invention is an actual list, which includes addresses of objects **to be deleted from memory**. The list is updated to include any objects that were not deleted within the calculated residual time, and stored so that the undeleted objects may be deleted in another communication cycle.

In Hasbun, there is no list of objects to be deleted from memory, and there is no updating and storing of the list. In the office action (page 4), the examiner states "selecting a block puts all dirty sectors on the block on the list of objects to be deleted". However, this is a complete mischaracterization of Hasbun. In Hasbun, no list is made; i.e., there is no actual list of objects to be deleted in Hasbun. The focus block of Hasbun is an actual block of data that includes sectors. FIG. 3 illustrates a block and the file structure of the block. This is not a list, and Hasbun never teaches making a list of objects to be deleted when one of these blocks is designated.

Further, claim 1 states that the list of the present invention is updated and stored. However, there is no list that is updated and stored in Hasbun. The only time Hasbun speaks of updating is when Hasbun updates databases 93 and sector header translation table 94. It is clear that these are not lists of objects to be deleted, and instead include various types of information on both valid and invalid data. It is also clear that these lists are not updated in the manner claimed in claim 1.

Clean up in Hasbun is done by a clean-up state machine (CSM). Each state of the CSM of Hasbun is designated by a bubble shown in FIG. 7. In Hasbun, when selecting a block to clean-up, the entire block is selected for deletion. This is because flash devices must be erased an entire block at a time (Hasbun col. 3, lines 61-63). However, no actual list is made of the dirty sectors in the block that is to be deleted.

Valid sectors as well as dirty sectors exist in Hasbun. The valid sectors are relocated in states 383-386 shown in FIG. 7. In Hasbun, erasure is not started until all valid sectors have been copied (Hasbun FIG. 7 and col. 11, lines 28-35). Erasure of the entire block then begins (Hasbun col. 10, lines 36-42).

It is clear from the above description, that in Hasbun there is no list of objects to be deleted. Instead, an entire block of data is being deleted after valid data is relocated. This can in no way be interpreted as a list.

Additionally, claim 1 of the present invention specifically recites that the list is updated --to list those undeleted objects of the first list after the lapse of the calculated residual time--, and that the updated list is stored such that the objects in the updated first list are available for deletion in another communication cycle--. This element is clearly not taught by Hasbun. Hasbun does not start its deletion process until all valid data has been relocated. Therefore, there is no reason for Hasbun to update --a list of objects to be deleted-- when an entire block is being deleted, and there is no reason for Hasbun to store the list when Hasbun is dealing with a single block all of which is being deleted. Further, Hasbun never mentions a list of objects to be deleted that is updated and stored.

The present invention states --***making*** a first list of objects to be deleted from a memory--. Nowhere does Hasbun mention making a list of objects to be deleted, and to state that in Hasbun, when a block is selected a list is made with the dirty sectors on the block would be a complete mischaracterization of Hasbun.

In the last full paragraph of the office action (page 4), the examiner states that in Hasbun, a block marked dirty is stored as a remaining object that has not been deleted

from memory. However, claim 1 specifically stated that a **list** was stored, and not merely an object. Also, the stored list listed remaining objects, which had not been deleted during the residual time. Claim 1 has been amended to further clarify the updated list of objects that were not deleted during the calculated residual time.

The examiner cites Hasbun col 8, lines 3-5 as teaching this aspect of the present invention. However, col. 8, lines 3-5 of Hasbun instead describes a write of a sector of data. In Flash memory, a sector of data is written to a new location each time the sector of data is revised rather than erasing the previous physical location and writing the revised sector of data in that same location (Hasbun col. 3 lines 21-25). As such, the previous version of the data is marked dirty and the sector header translation table is updated. As such, Hasbun col. 8, lines 3-5 is merely a situation where a write is occurring. This is not a situation where a --first list-- is updated --to list those undeleted objects-- which remain --after the lapse of the calculated residual time--. The applicants would also like to note, that in Hasbun, when a sector is marked dirty, it is not placed on any list, it is marked dirty within the attribute word shown in FIG. 3.

As described, in Hasbun, there is no list of objects to be deleted, and no updating and storing of the list of objects after the lapse of the calculated residual time. Accordingly, Hasbun does not teach the present invention of claim 1. An indication of allowable subject matter with respect to claim 1 is respectfully requested.

As to claims 10 and 13, the applicants respectfully submit that these claims are allowable for the same reasons described above. As such, the applicants would like to resubmit the above arguments.

As to claims 2-3, the applicants respectfully submit that these claims are allowable at least since they depend from claim 1, which is now considered to be in condition for allowance for the reasons above.

Claims 4-6 stand rejected under 35 U.S.C. § 103(a) as being obvious over Hasbun in view of U.S. Patent No. 5,355,483 (Serlet).

The applicants respectfully submit that these claims are allowable at least since they depend from claim 1. More specifically, nothing in Hasbun or Serlet teaches --making a first list of objects to be deleted--, --updating-- the first list-- to list --undeleted objects of the first list which remain after the lapse of the calculated residual time--, and --storing the updated first list--.

Claims 7-9 and 11-12 stand rejected under 35 U.S.C. § 103(a) as being obvious over Hasbun in view of U.S. Patent No. 5,740,395 (Wells).

The applicants respectfully submit that these claims are allowable at least since they depend from claim 1. More specifically, nothing in Hasbun, Serlet, or Wells teaches --making a first list of objects to be deleted--, --updating-- the first list-- to list --undeleted objects of the first list which remain after the lapse of the calculated residual time--, and --storing the updated first list--.

For the reasons set forth above, the applicants respectfully submit that claims 1-13 pending in this application, are in condition for allowance over the cited references. Accordingly, the applicants respectfully request reconsideration and withdrawal of the outstanding rejections and earnestly solicit an indication of allowable subject matter.

This amendment is considered to be responsive to all points raised in the office

action. Should the examiner have any remaining questions or concerns, the examiner is encouraged to contact the undersigned attorney by telephone to expeditiously resolve such concerns.

Respectfully submitted,

Dated: March 5, 2008

W. William Park, Reg. No. 55,523
Ladas & Parry LLP
224 South Michigan Avenue
Chicago, Illinois 60604
(312) 427-1300

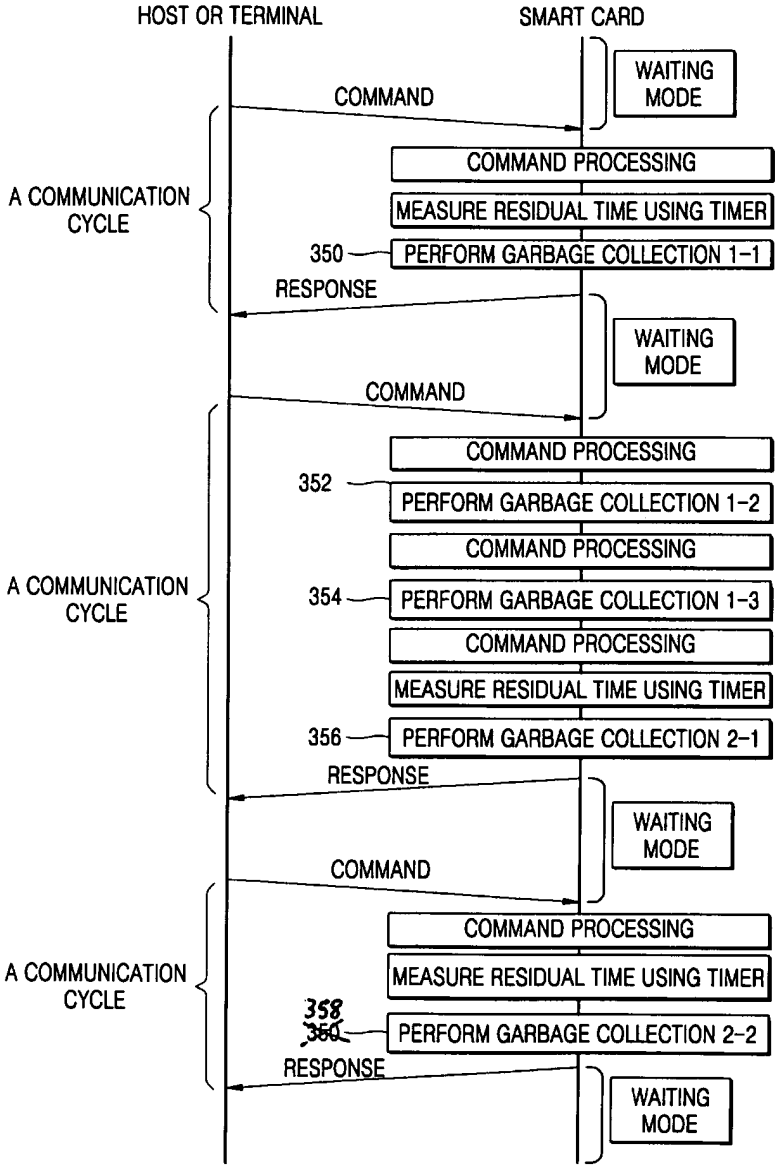
APPENDIX OF ATTACHMENTS

Replacement Sheet of FIG. 3F
(a total of ONE drawing sheet)

and

Annotated Sheet Showing Changes of FIG. 3F
(a total of ONE drawing sheet)

FIG. 3F


HOST OR TERMINAL
SMART CARD
WAITING MODE
COMMAND
COMMAND PROCESSING
MEASURE RESIDUAL TIME USING TIMER
A COMMUNICATION CYCLE
350
PERFORM GARBAGE COLLECTION 1-1
RESPONSE
WAITING MODE
COMMAND
COMMAND PROCESSING
352
PERFORM GARBAGE COLLECTION 1-2
COMMAND PROCESSING
A COMMUNICATION CYCLE
354
PERFORM GARBAGE COLLECTION 1-3
COMMAND PROCESSING
MEASURE RESIDUAL TIME USING TIMER
356
PERFORM GARBAGE COLLECTION 2-1
RESPONSE
WAITING MODE
COMMAND
COMMAND PROCESSING
MEASURE RESIDUAL TIME USING TIMER
A COMMUNICATION CYCLE
358
360
PERFORM GARBAGE COLLECTION 2-2
RESPONSE
WAITING MODE